HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY OF INFORMATION TECHNOLOGY

FINAL TERM PROJECT

Course name: Algorithms And Data Structures

HOTEL MANAGEMENT PROGRAMS

Lecturer name: Assoc. Prof. Hoang Van Dung

Group 7:
Member: Nguyễn Đình Hồng Quân 19119045
Trần Phan Bảo Khang 19119059
Nguyễn Trọng Hải 19119028

Ho Chi Minh City, 06/2022

HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY OF INFORMATION TECHNOLOGY

FINAL TERM PROJECT

Course name: Algorithms And Data Structures

HOTEL MANAGEMENT PROGRAMS

Lecturer name: Assoc. Prof. Hoang Van Dung
Group 7:
Member: Nguyễn Đình Hồng Quân 19119045
Trần Phan Bảo Khang 19119059
Nguyễn Trọng Hải 19119028

Ho Chi Minh City, 06/2022

STUDENT PROCESS

ORDER	NAME	STUDENT ID	Contribution rate
1	Nguyen Dinh Hong Quan	19119045	100%
2	Tran Phan Bao Khang	19119059	100%
3	Nguyen Trong Hai	19119028	100%

Comment of Lecture:

……
……
……
……
……
……
……
……
……
……

Ho Chi Minh, day…..June, 2022

Score

(Sign and full name)

Hoang Van Dung

Task assignment

Nguyen Trong Hai	Tran Phan Bao Khang	Nguyen Dinh Hong Quan
Prepare word, and ppt reports, write programming code on menu building part, in which research on editing customer information.	Prepare ppt, write programming code about generating room number corresponding to room category, give a reasonable price for each room, store customer information.	Prepare word, and ppt reports, write programming code for menu building, in which research on deleting customer information, statistics on the number of rooms booked by customers

Acknowledgment

We would like to express our sincere thanks to Dung’teacher who guided us in the preparation of this report. He gave many ideas for us to grasp the problem of our report and besides, the presentation will be more effective. Our report has been successful thanks to Mr. Dung's guidance over the past three weeks.

However, due to many problems that arise as well as the time we do it, there are still many mistakes that are inevitable. We look forward to receiving all of your comments and suggestions to help our problems be answered and improved.

Sincerely thanks.

Preface

The purpose and objective of this course and mainly the content are to know data structure (ex: Array, data type struct, class, Stack, Queue, tree...) and Algorithms(sort, search, add, modify, delete).

We have gained more confidence regarding coding and introducing products. We also believe that will make our group gain some kind of IT knowledge, and if we practice much and much will having some experiments in the future, then we will be able to survive smartly in today's competitive environment.

The effort to write the report is a partial process to complete the course. In the report, we try to divide each of the topics into an individual chapter to reflect the topic more clearly.

Finally, we are very hopeful our topic of the report and code will be useful material for all the readers and users.

List of images

Figure 1. Block diagram....12
Figure 2. Header file....13
Figure 3. Public room....13
Figure 4. Pulic hotel....13
Figure 5. Struct room....13
Figure 6. Struct customer....14
Figure 7. Room information....14
Figure 8. Customer information....14
Figure 9. Add room....15
Figure 9.1.Information Standard Room....15
Figure 9.2.Information Moderate Room....15
Figure 9.3.Information Superior Room....16
Figure 9.4.Information Junior Room....16
Figure 9.5.Information Suite Room....16
Figure 10. Search Room....17
Figure 11. Delete Room....17
Figure 12. Check room_no....18
Figure 13. Check room_status....18
Figure 14. Set status....18
Figure 15. Check-in....19
Figure 16. Check-out....19
Figure 17. Get Available Room....20
Figure 18. Search customer....20
Figure 19. Guest summary report....21
Figure 20. Manage room....21
Figure 21. Menu....22
Figure 22. Test Menu....23
Figure 22.1. Test add room....23
Figure 22.2. Test search room....24
Figure 22.3.Test delete room....25
Figure 23. Test check-in room....26
Figure 24. Test available rooms....26
Figure 25. Test search customer by name....27
Figure 26. Test check-out....27
Figure 27. Test summary....28
Figure 28. Exit....28

CONTENT

Acknowledgment 5
Preface..... 6
List of images..... 7
I..... Project description.....9
1... Objectives..... 9
2... Scope and object.....9
II..... Theoretical basis.....10
1.... Dev-C++..... 10
2.... C++ language.....10
3.... Linked list..... 10
4.... Deployment method..... 10
III.....Design.....12
1... Block diagram..... 12
2... Process description..... 12
3... The project's main features.....12
4... Implementation of Hotel Management System..... 12
IV..... Test Cases.....23
V.....Conclusion.....29
1... Result..... 29
2... Difficulties and limitations 29
3... Development of ideas.....29
References..... 30

I..... Project description

1... Objectives

With the topic "Building a hotel management program", our main goal is to build a program to support effective hotel management and can be applied in practice, managing booking problems. room, edit information, check-out, room type, invoice.

The program has a simple interface, effective management, easy to use.

2... Scope and object

Project on building a hotel management system. That shows just based on the hotel's booking and payment management. Users can manage bookings and, customer profiles, view total allocated rooms, edit infor,mation, and make payments.

II..... Theoretical basis

1... Dev-C++

Dev-C++ is a full-featured integrated development environment (IDE), which is able to create Windows or DOS-based C/C++ programs using the Mingw compiler system (included with the package), or the Cygwin compiler. These are the recommended requirements of Dev-C++:

+Microsoft Windows 98, NT or 2000

+32 MB RAM

+233 Mhz Intel-compatible CPU

+45 MB free disk space

Dev-C++ allows you to write, compile and run a C or C++ program. C++ programming language is an enhanced version of the C language that provides object-oriented programming (OOP) capabilities. It is a superset of C, which means that you can use a C++ compiler to compile C programs. Object-oriented programming techniques differ significantly from the sequential programming used in the C programming language. Although a C++ compiler like Dev-C++ allows you to compile a C program that includes some features of C++, in this course, we will concentrate on the C programming language. A program written in pure C language may be compiled and run using other C compilers, like Turbo C, etc.

2... C++ language

C++ is a high-level computer programming language. Developed by Bjarne Stroustrup of Bell Laboratories in the early 1980s, it is based on the traditional C language but with added object-oriented programming and other capabilities. C++, along with Java, has become popular for developing commercial software packages that incorporate multiple interrelated applications. Large parts of many operating systems are written in the language. C++ is considered one of the fastest languages and is very close to low-level languages, thus allowing complete control over memory allocation and management. This very feature and its many other capabilities also make it one of the most difficult languages to learn and handle on a large scale.

3... Linked List

A linked list is a data structure used to store a set of discrete elements that can be dynamically expanded.The size of the linked list does not need to be defined in advance, it automatically changes when the number of elements in the list changes.

+ Unlimited number of elements

+ Easy to perform operations: add, delete, edit

+ Sequential data retrieval

4... Deployment method

+ Learn about the room management mechanism

+Learn C++ programming language

+ Dev-C++ support tools
+ Analyze functions
+Database Design
+Writing Programs
+ Implement and evaluate the results

III. Design

1. Block diagram

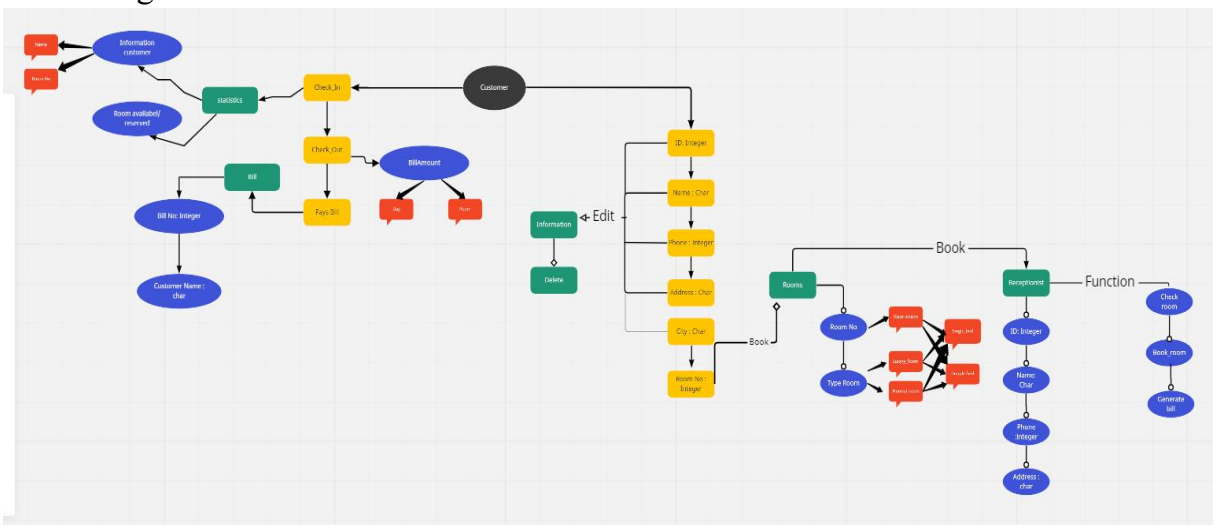


Figure 1.Block diagram

2. Process description

   It includes class, object, array, function, loop, and structure in hotel management projects based on C++ language. When this code is launched, the user will see a menu with five options numbered from 1 to 5, then the program will execute the correct choice of the user to output the functions that match the content needed to capture information.

3. The project's main features

   - Room management: In this project, we can manage and statistics the number of rooms that are available or booked.
   - Guest Check-in and Check-out: Using this project, we can track guest check-in or check-out activity over time.
   - Check-in availability: we make a list of available rooms
   - Edit customer information: Adjust customer information when incorrect information is entered
   - Searching for customers: Using this feature, we can search for any customer, what type of room is in and how many rooms are there
   - Invoice printing: this feature helps to calculate the amount the customer needs to pay

4. Implementation of Hotel Management System

Step1: Create a new **Project** in Dev C++

File > New > Source File or Ctrl + N

Step2: we will include all **header file** which will use in project

```
#include<iostream>
#include<conio.h>
#include<fstream>
```

Figure 2.Header file

Step3 : Add the following line below **header** file:

```
using namespace std;
```

ROOM

Step4: we will create a class(**room**) that can take the details of the room through a public

```
class Room
{
public:
  int roomid;
  string name;
  string address;
  string phone;
  string fromdate;
  string todate;
  float paymentinadv;
  string bookingid;
  void insert1(string,string,string,string,string,float,string,int);
  void addRoom(int);
  void searchRoom(int);
  void Delete(int);
```

Figure 3.Public room

Step5: we will create a public which can take the details of the menu hotel.

```
ofstream exportfile ("dataexport.txt");
  time_t now = time(0);
  char* dt = ctime(&now);
```

Figure 4. Pulic hotel

Step6: we will create a struct which can take the details of the room

```
struct Stroom
{
    string ac;
    string type;
    string stype;
    int rent;
  int roomnumber;
  int status;
  struct Stroom *next;
}*head;
```

Figure 5. Struct room

Step7: we will create a struct which can take the details of the customer

```
struct Stcustom
{
   int roomid;
   string name;
   string address;
   string phone;
   string fromdate;
   string todate;
   float paymentinadv;
   string bookingid;
   struct Stcustom *next;
}*headcustom;
```

Figure 6. Struct customer

Step8: The insert function is to insert a node into the single list of Room

```
void insert(string ac1, string type1, string stype1, int rent1, int roomnumber1,int status1)
{
    struct Stroom *stobj = new Stroom ;
    stobj->ac=ac1;
    stobj->type=type1;
    stobj->stype=stype1;
    stobj->rent=rent1;
    stobj->roomnumber=roomnumber1;
    stobj->status=status1;
    stobj->next=NULL;

    if(head==NULL){
        head=stobj;
    }
    else{
        stobj->next=head;
        head=stobj;
    }

}
```

Figure 7.Room information

Step9: The insert 1 function is to insert a node into the customer's single list

```
void Room::insert1(string name1, string address1, string phone1,string fromdate1, string todate1, float paymentinadv1, string bookingid1,int roomid1 )
{

    struct Stcustom *stobj = new Stcustom ;

    stobj->name=name1;
    stobj->address=address1;
    stobj->phone=phone1;
    stobj->fromdate=fromdate1;
    stobj->todate=todate1;
    stobj->paymentinadv=paymentinadv1;
    stobj->bookingid=bookingid1;
    stobj->roomid=roomid1;
    stobj->next=NULL;

    if(headcustom==NULL)
    {
        headcustom=stobj;
    }
    else{
        stobj->next=headcustom;
        headcustom=stobj;
    }

}
```

Figure 8. Customer information

Step 6 : Now we will create a function to add, search, delete

- Add Room

```
void Room::addRoom(int rno)
{
  string ac;
  string type;
  string stype;
  int rent;
  int opt;
  system("cls");
  cout<<"
      cout<<"
      cout<<"
      cout<<"
      cout<<"
      cout<<"
  cout<<"
  cout<<"
  cin>>opt;
  exportfile<<"Data of Room Number : "<<rno<<endl;
  switch(opt)
```

```
____Enter type of Rooms_____\n";
    | 1. Standard               |\n";
    | 2. Moderate               |\n";
    | 3. superior               |\n";
    | 4. Junior suite           |\n";
    | 5. Suite                  |\n";
-----------------------------\n";
Enter Option:\n";
```

Figure 9. Add room

In function add room we have Room type, Comfort type, Type Size, Room Rent

```
_____Standard Room info_____\n";
| Room Type Non-AC                        |\n";
| Type Comfort - Single bed               |\n";
| Type Size - Small                       |\n";
| Daily Rent is Rs 300 / 24 hours         |\n";
| Room  information Added Successfully!   |\n";
-------------------------------------------\n";
```

Figure 9.1.Information Standard Room

```
_____Moderate Room info_____\n";
| Room Type AC                            |\n";
| Type Comfort - Single bed               |\n";
| Type Size - Medium                      |\n";
| Daily Rent is Rs 500 / 24 hours         |\n";
| Room  information Added Successfully!   |\n";
-------------------------------------------\n";
```

Figure 9.2.Information Moderate Room

```
_____Superior Room info_____\n";
| Room Type AC                |\n";
| Type Comfort-Double bed + living room |\n";
| Type Size - Big             |\n";
| Daily Rent is Rs 1000 / 24 hours |\n";
| Room  information Added Successfully! |\n";
-------------------------------\n";
```

Figure 9.3.Information Superior Room

```
_____Junior suite Room info_____\n";
| Room Type AC                |\n";
| Type Comfort-Doublebed+lakeview+living|\n";
| Type Size - Big             |\n";
| Daily Rent is Rs 2000 / 24 hours |\n";
| Room  information Added Successfully! |\n";
-------------------------------\n";
```

Figure 9.4.Information Junior Room

```
_____Suite Room info_____\n";
| Room Type AC                |\n";
| Type Comfort-2bedrooms+lakeview+living|\n";
| Type Size - Big             |\n";
| Daily Rent is Rs 5000 / 24 hours |\n";
| Room  information Added Successfully! |\n";
-------------------------------\n";
```

Figure 9.5.Information Suite Room

❖ Search Room

```
void Room::searchRoom(int rno)
{
    int flag=0;
    system("cls");
    struct Stroom *temp=head;
    while(temp!=NULL){
        if(temp->roomnumber==rno)
        {
            cout<<"Room Number : "<<temp->roomnumber<<endl;
            cout<<"Room Type : "<<temp->ac<<endl;
            cout<<"Type comfort : "<<temp->type<<endl;
            cout<<"type size : "<<temp->stype<<endl;
            cout<<"rent : "<<temp->rent<<endl;

        flag=1;
        }
        temp = temp->next;
    }
    if(flag==0)
    {
            cout<<rno<<" is not available ";
    }
```

Figure 10. Search Room

❖ Delete Room

```
void Room ::Delete(int rno)
{
  int flag=0;
  struct Stroom * temp1 ,*temp2;
  temp1 = temp2 = head;

  while(temp1!=NULL)
  {
    if(temp1->roomnumber==rno)
    {
        cout<<"Record against this room is available \n";
        if(temp1==temp2)
        {
        head = head->next;
        delete(temp1);
        }
        else{
        temp2->next = temp1->next;
        delete(temp1);
        }
        flag=1;
        cout<<"Record Against Room Number "<<rno<<" is deleted\n";
    }
    temp2 = temp1;
    temp1 = temp1->next;
  }
```

Figure 11. Delete Room

Checkroom is to go through the single list to check if the room number is created in the single list. If yes, return true.

```
bool checkroomno(int rno)
{
  struct Stroom *temp=head;
  while(temp!=NULL)
  {
    if(temp->roomnumber==rno )
    {
      return true;
    }
    temp = temp->next;
  }
}
```

Figure 12. Check room_no

Check Room Status is to check if the room number has been booked

```
bool checkroomstatus(int rno)
{
  struct Stroom *temp=head;
  while(temp!=NULL)
  {
    if(temp->roomnumber==rno )
    {
      if(temp->status==1)
      {
        return true;
      }
    }
    temp = temp->next;
  }
  return false;
}
```

Figure 13. Check room_status

Setstatus is to update the room number that is already booked

```
void setstatus(int rno,int stat)
{
  struct Stroom *temp=head;
  while(temp!=NULL)
  {
    if(temp->roomnumber==rno)
    {
      temp->status=stat;
      return;
    }
    temp = temp->next;
  }
}
```

Figure 14. Set status

HOTEL MANAGE

In hotel management we have function: a check-in, check out, get available, search for customer and guest summary, manage room.

❖ Check-in

```
void hotelmanage::checkIn()
{
  int found=0,rno;
  cout<<"\nEnter Room number : ";
  cin>>rno;
  if(checkroomno(rno)==true)
  {
    found=1;
  }
  else{
    cout<<"you need to add room first \n";
    getch();
  }
  if(found==1)
  {
    if(checkroomstatus(rno)==true)
    {
      cout<<"\nRoom is already Booked";
      getch();
      return;
    }
    struct Stcustom *temp=headcustom;
    cout<<"\nEnter booking id: ";
```

Figure 15. Check-in

❖ Check-out

```
                    Checkout Details                    \n";
|Customer Name : "<<tempcust->name<<"\n";
|Room Number : "<<tempcust->roomid<<"\n";
|Address : "<<tempcust->address<<"\n";
|Phone : "<<tempcust->phone<<"\n";
|Total Amount Due : "<<billAmount<<"\n";
|Advance Paid: "<<tempcust->paymentinadv<<"\n";
|Total Payable: "<<billAmount-tempcust->paymentinadv<<"only"<<"\n";
-------------------------------------------------------------------\n";
```

Figure 16. Check-out

❖ Get available

```
void hotelmanage::getAvailRoom()
{
  int found=0;
  struct Stroom *temp = head;
  while(temp!=NULL)
  {
    if(temp->status==0)
    {
       cout<<"Room no "<<temp->roomnumber<<" is Available \n";
       found=1;
    }
    temp = temp->next;
  }
  getch();
  if(found==0)
  {
    cout<<"\nAll rooms are reserved";
    getch();
  }
}
```

Figure 17. Get Availble Room

- Search customer

```
void hotelmanage::searchCustomer(string pname)
{
   int i,found=0;
   struct Stcustom *tempcust = headcustom;
   while(tempcust!=NULL)
   {
     if(tempcust->name==pname)
     {
       cout<<"\nCustomer Name: "<<tempcust->name;
       cout<<"\nRoom Number: "<<tempcust->roomid;
       cout<<"\n\n Record desplayed Press enter for next record";
       found=1;
       getch();
     }
     tempcust=tempcust->next;
   }
   if(found==0)
   {
     cout<<"\nPerson not found.";
     getch();
   }
}
```

Figure 18. Search Customer

❖ Guest summary

```
void hotelmanage::guestSummaryReport()
{
  struct Stcustom *temp = headcustom;
  while(temp!=NULL)
  {
    cout<<"\n Customer First Name : "<<temp->name;
    cout<<"\n Address (only city) : "<<temp->address;
    cout<<"\n Booking ID          : "<<temp->bookingid;
    cout<<"\n Check in date       : "<<temp->fromdate;
    cout<<"\n Checkout date       : "<<temp->todate;
    cout<<"\n Occupied Room No    : "<<temp->roomid;
    cout<<"\n Phone               : "<<temp->phone;
    cout<<"\n---------------------------------------";
    temp= temp->next;
  }
  getch();
}
```

Figure 19. Guest summary report

❖ Manage room

```
void manageRooms()
{
  class Room room;
  int opt,rno,flag=0;
  do
  {
    system("cls");
    cout<<"                                        _______Manage rooms_______\n";
    cout<<"                                       | 1. Add Rooms              |\n";
    cout<<"                                       | 2. Search Room            |\n";
    cout<<"                                       | 3. Delete Room            |\n";
    cout<<"                                       | 4. Back to main menu      |\n";
    cout<<"                                       ----------------------------\n";
    cout<<"                                       Enter Option:\n";
    cin>>opt;
    switch(opt)
    {
      case 1:
        cout<<"\nEnter Room Number: ";
        cin>>rno;
        if(checkroomno(rno)==true)
        {
            flag=1;
        }
        if(flag==1)
        {
          cout<<"\nRoom Number is already Present\n ";
          cout<<"Please enter Different Number";
```

Figure 20. Manage room

Final step is Menu

```
int main()
{
  class hotelmanage hm;
  int opt,rno;
  char ch;
  string sname;
  system("cls");
  do
  {
    system("cls");
    cout<<"                                          Designed By - Nguyen Trong Hai - 19119028\n";
    cout<<"                                                      - Tran Phan Bao Khang - 19119028\n";
    cout<<"                                                      - Nguyen Dinh Hong Quan - 19119045\n";
    cout<<"                                         ----------------------------------------------\n";
    cout<<"                                             _____ Hotel Management _____\n";
    cout<<"                                           | 1. Manage Rooms             |\n";
    cout<<"                                           | 2. Check-In Room            |\n";
    cout<<"                                           | 3. Available Rooms          |\n";
    cout<<"                                           | 4. Search Customer By Name |\n";
    cout<<"                                           | 5. Check-Out Room           |\n";
    cout<<"                                           | 6. Summary                  |\n";
    cout<<"                                           | 7. Exit                     |\n";
    cout<<"                                           -------------------------------\n";
    cout<<"                                           Enter Option:\n";
    cin>>opt;
```

Figure 21. Menu

IV. Test case

❖ Menu

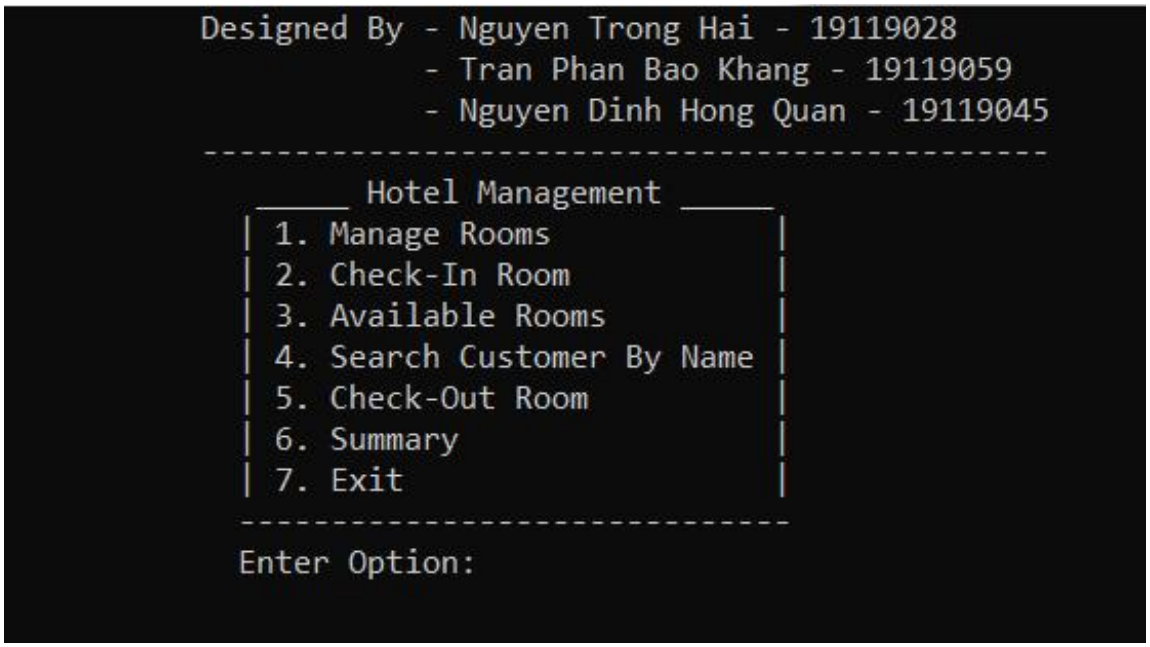


Figure 22.Test Menu

Case1 : Manage room

- Test Add room

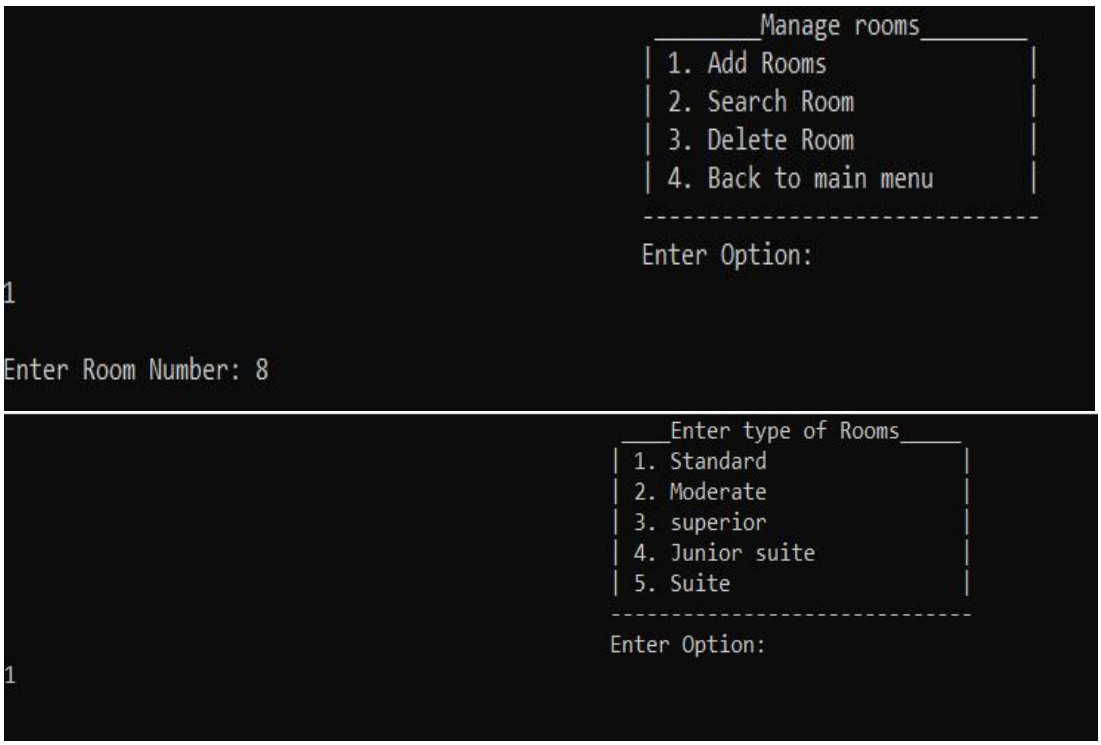


Figure 22.1. Test add room

After create room , it will show infomation of type of room

```
          Standard Room info
    | Room Type Non-AC                        |
    | Type Comfort - Single bed               |
    | Type Size - Small                       |
    | Daily Rent is Rs 300 / 24 hours         |
    | Room  information Added Successfully!   |
    -------------------------------------------
```

- Test Search Room

```
                                        Manage rooms
                                    | 1. Add Rooms                |
                                    | 2. Search Room              |
                                    | 3. Delete Room              |
                                    | 4. Back to main menu        |
                                    -------------------------------
                                    Enter Option:
2

Enter room number: 8
Room Number : 8
Room Type : Non-AC
Type comfort : Single bed
type size : small
rent : 300
```

Figure 22.2. Test search room

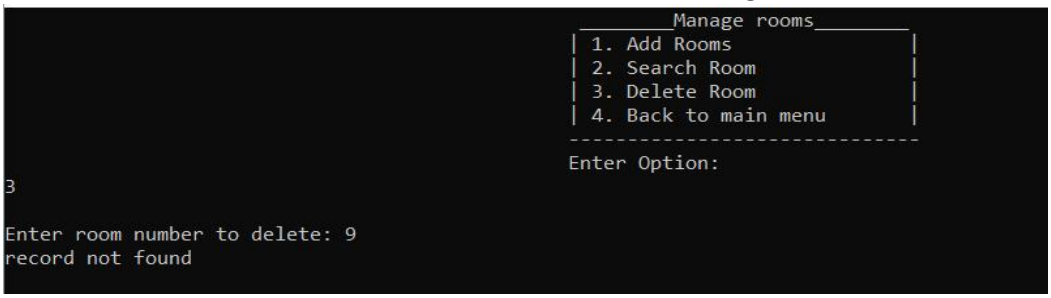
If input room wrong it will show

```
                                        Manage rooms
                                    | 1. Add Rooms                |
                                    | 2. Search Room              |
                                    | 3. Delete Room              |
                                    | 4. Back to main menu        |
                                    -------------------------------
                                    Enter Option:
2

Enter room number: 9
9 is not available
```

- Test Delete Room

Delete Room is not available it will show text in figure below



If input correct, it will delete

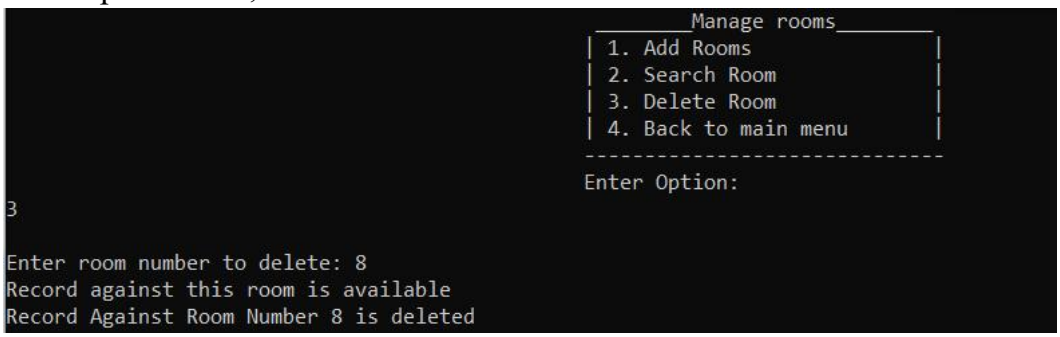
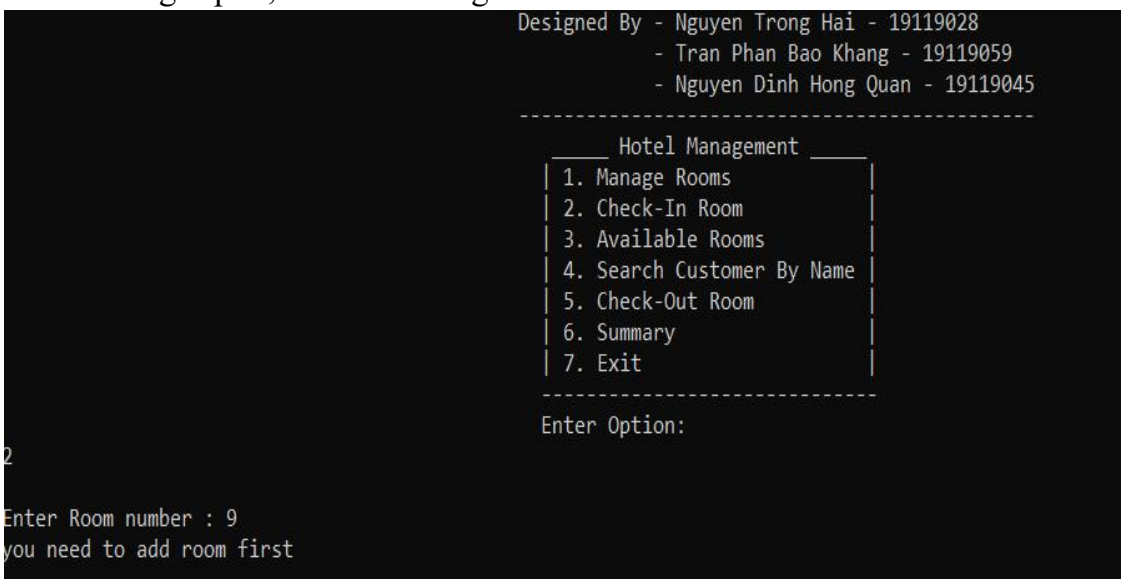


Figure 22.3.Test delete room

Case 2 : Check-in Room

Wrong Input , it will warning



Right Input , we will fill information

```
                                          | 1. Manage Rooms             |
                                          | 2. Check-In Room            |
                                          | 3. Available Rooms          |
                                          | 4. Search Customer By Name  |
                                          | 5. Check-Out Room           |
                                          | 6. Summary                  |
                                          | 7. Exit                     |
                                          -------------------------------
                                          Enter Option:
2

Enter Room number : 8

Enter booking id: 19119059

Enter Customer Name (First Name): Khang

Enter Address (only city): DongNai

Enter Phone: 09084128100

Check-in Date and Time : Wed Jun 01 09:24:19 2022

Enter to  Date: 1/6/2022

Enter Advance Payment: 20

 Customer Checked-in Successfully..
```

Figure 23.Test check-in room

Case 3 : Available Rooms

```
                                  Designed By - Nguyen Trong Hai - 19119028
                                              - Tran Phan Bao Khang - 19119059
                                              - Nguyen Dinh Hong Quan - 19119045
                                 ------------------------------------------
                                     _____ Hotel Management _____
                                   | 1. Manage Rooms             |
                                   | 2. Check-In Room            |
                                   | 3. Available Rooms          |
                                   | 4. Search Customer By Name  |
                                   | 5. Check-Out Room           |
                                   | 6. Summary                  |
                                   | 7. Exit                     |
                                   -------------------------------
                                   Enter Option:
3
Room no 8 is Available
```

Figure 24. Test availble rooms

Case 4 : Search Customer By Name

Name record is valid

```
                                  Designed By - Nguyen Trong Hai - 19119028
                                              - Tran Phan Bao Khang - 19119059
                                              - Nguyen Dinh Hong Quan - 19119045
                                 ------------------------------------------
                                     _____ Hotel Management _____
                                   | 1. Manage Rooms             |
                                   | 2. Check-In Room            |
                                   | 3. Available Rooms          |
                                   | 4. Search Customer By Name  |
                                   | 5. Check-Out Room           |
                                   | 6. Summary                  |
                                   | 7. Exit                     |
                                   -------------------------------
                                   Enter Option:
4
Enter Customer Name: Khang

Customer Name: Khang
Room Number: 8

 Record desplayed Press enter for next record
```

Name record is wrong

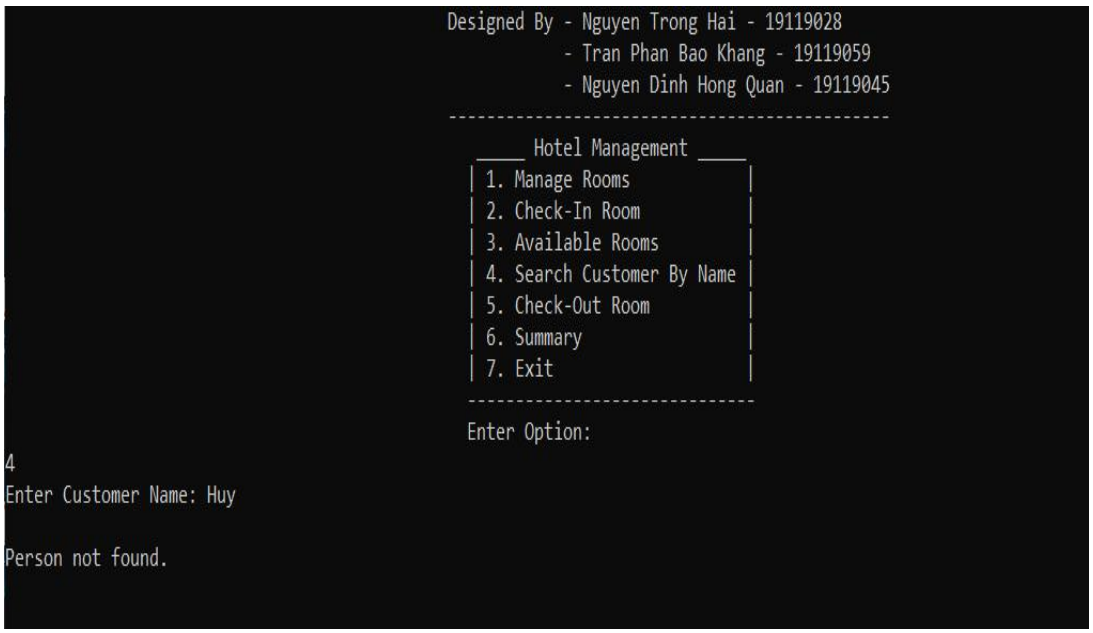


Figure 25. Test search customer by name

Case 5 : Check- out Rooms

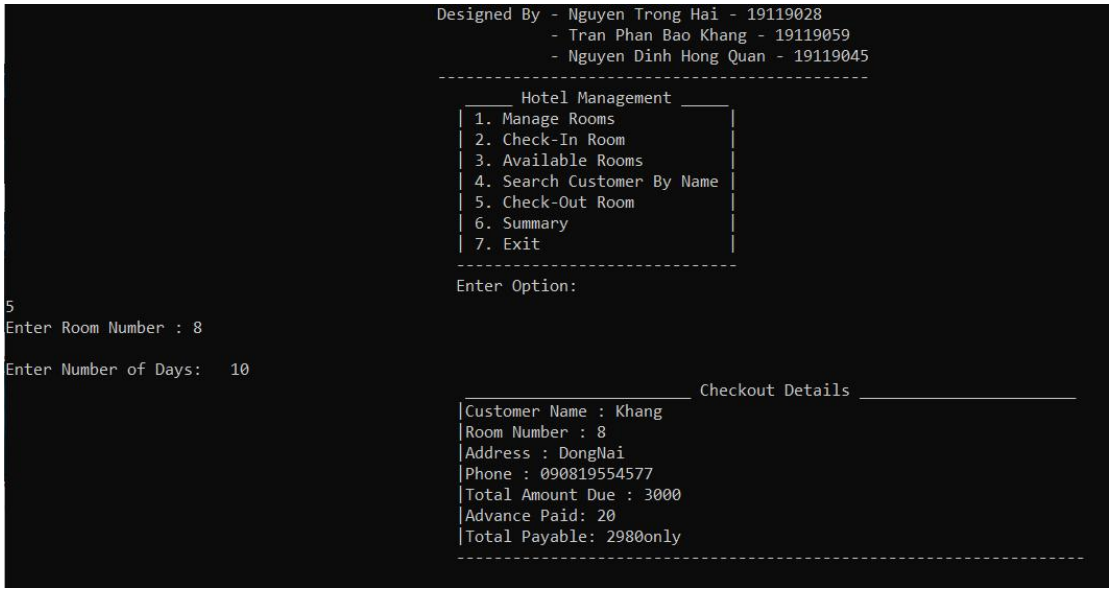


Figure 26. Test check-out

Case 6 : Summary

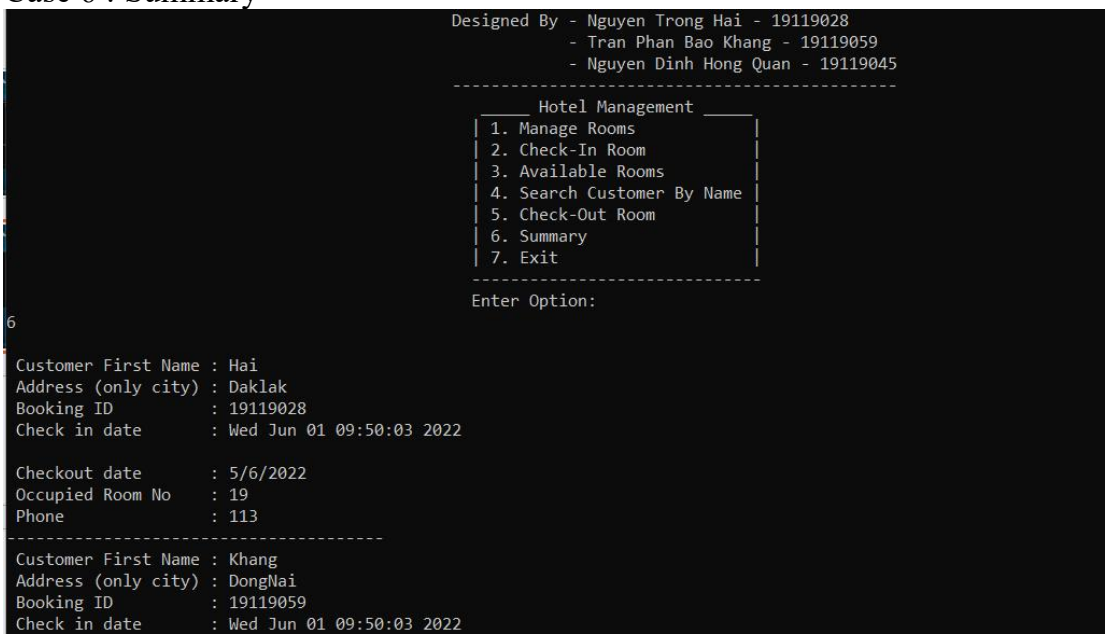


Figure 27. Test summary

Case 7 : Exit

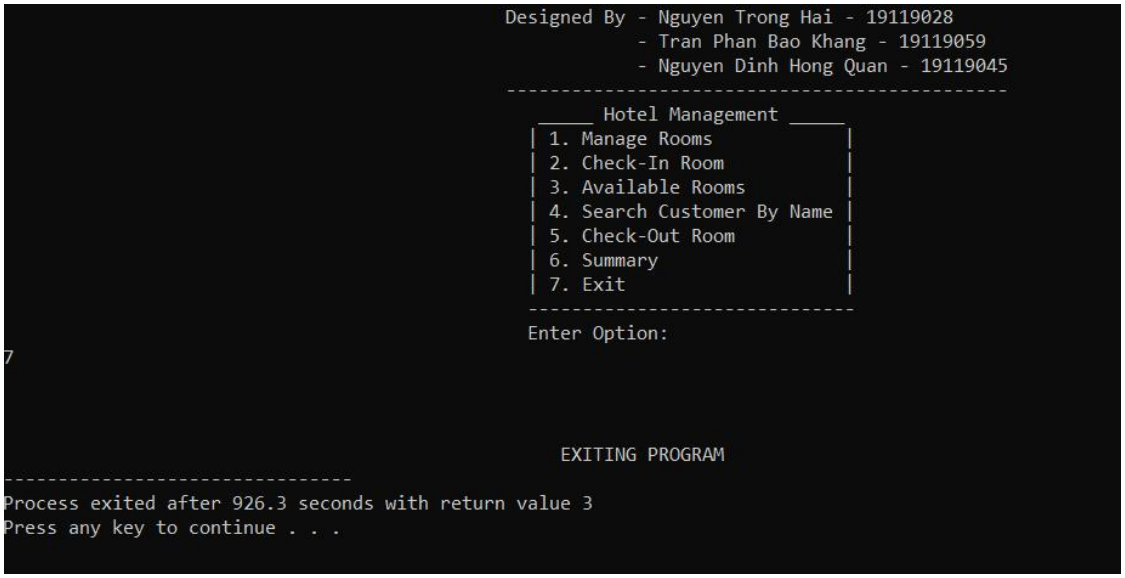


Figure 28. Exit

V. Conclusion

1. Result

   Successfully implemented the construction of a hotel management system in C++ language, displaying information through the screen, selecting and implementing, implementing,….

2. Difficulties and limitations

   Customer satisfaction has not been assessed yet

3. Development of ideas

   Will do more management work such as assessing guest satisfaction, allowing to assess whether the service level of the hotel is good or not.

References

1. https://www.academia.edu/43499326/Hotel_Management_Project_Using_C_OOP_structure

2. https://rrtutors.com/tutorials/hotel-management-system-project-using-c-plus-plus

3. https://www.cppbuzz.com/projects/c++/c++-project-on-hotel-management

Link video nhóm :
https://drive.google.com/drive/folders/1n1j8rhp9mpEiiDg262sOZ5zsMd02JBGX?usp=sharing